Cleanse Utilization Process

Expected files:

Name of files (as received)	File Type	Primary Identifier(s)	Mandatory
demog	Demographic Information	sourcePersonId	Mandatory
visit	Visit (Activity) related Information	sourcePersonId, sourceRecordId, primaryCarePhysicianId	Mandatory
cpt	Current Procedural Terminology	sourcePersonId, sourceRecordId	Mandatory
px	Prognosis	sourcePersonId, sourceRecordId	Mandatory
dx	Diagnosis	sourcePersonId, sourceRecordId	Mandatory
facility	Hospital/ Business/ Clinic/ Site/ Practice Information	sourcePersonId, sourceRecordId	Mandatory
financial	Financial Information	sourcePersonId, sourceRecordId	Not Mandatory
biometric	Biometric Information	sourcePersonId, sourceRecordId	Not Mandatory
physician	Physician Information	primaryCarePhysicianId	Not Mandatory
guarantor	Guarantor Information	sourcePersonId, sourceRecordId	Not Mandatory

Normalization Process:

Steps	Description	Join Type	Join Key	Remarks
1	Reads all the input files, checks for mandatory files and filters header rows if they exist.			
2	Join demog files with visit files = **joinedDemogAndVisitDf**	Full Outer	sourcePersonId	Results in cross product of all records from demog files and visit files. This includes: • common sourcePersonids, present in both *demog* and *visit* • sourcePersonids present in *demog* but not in *visit* • sourcePersonids present in *visit* but not in *demog*
3	Join files cpt, px, dx, facility, financial, biometric, guarantor with **joinedDemogAndVisitDf** = **joinedWithOtherDfExceptPhysician** Adding *dateCreated* column with current date to **joinedWithOtherDfExceptPhysician**	Left Outer on **joinedDemogAndVisitDf**	sourcePersonId, sourceRecordId	Using Reduce operation to join all dataframes
4	Join physician with **joinedWithOtherDfExceptPhysician** = **NormalizedDataFrame**	Left Outer on **joinedWithOtherDfExceptPhysician**	primaryCarePhysicianId	

Cleansing Process:

Steps	Description	Cleansing Columns	Cleansing Criteria
1	Splits dataframe into hasActivityDf and doesNotHaveActivity		hasActivityDf (containing sourceRecordId is not null) and doesNotHaveActivity (containing sourceRecordId is null)
2	Validate codes on hasActivityDf.	sourceExclusionFlag, sourcePatientType, sourceErPatient	sourceExclusionFlag: check for values 'Y', 'N' sourcePatientType: check for values 'I', 'O' sourceErPatient: check for values 'E', 'N'

3	Checking columns for null values on hasActivityDf & Spliting dataframe into activitiesContaingNulls & activitiesNotContainingNulls	firstName, lastName, address1, sourceSex, dateOfBirth, sourceType, source, hospitalId, hospital	activitiesContaingNulls has null values for cleansing columns and activitiesNotContainingNulls has non null values for cleansing columns
4	Format Dates for activitiesNotContainingNulls and Union doesNotHaveActivity dataframe = cleansedDf.	dateOfBirth, dateOfDeath, admitDate,dischargeDate, lengthOfStay	All dates are formatted to 'yyyy-MM-dd' format. Validation check, date is valid, date of death can't be before date of birth, dischargeDate can't be before admitDate
5	Format String columns on cleansedDf.		
6	Format Amount columns on cleansedDf.	visitTotalCharges, charges, cost, revenue, contributionMargin, profit	
7	Validate contact details on cleansedDf.	homePhone, mobilePhone, workPhone, email	google phone validation, email address validation
8	Adding zip5 and zip4 columns on cleansedDf.	zip5, zip4	zip splits into zip5 and zip4 columns

9	adding alias to extra columns on cleansedDf.	AddressType, ActivityType, sourceActivityType, activityDate, dischargeDate, customer, customerId	AddressType = 'HOME',
			ActivityType='ENCOUNTER',
			sourceActivityType = 'ENCOUNTER',
			activityDate = dischargeDate

10	Save *cleansed data* (cleansedDf) in parquet format, *error data* (activitiesContaingNulls) in csv format & *archive raw files*		